UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/747,091 | 12/21/2000 | Jeffrey L. Kodosky | 5150-45900 | 1583 |

35690 7590 06/06/2006

MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.
700 LAVACA, SUITE 800
AUSTIN, TX 78701

| EXAMINER | |
|---|---|
| KANG, INSUN | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2193 | |

DATE MAILED: 06/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/747,091 | KODOSKY ET AL. | |
| | Examiner | Art Unit | |
| | Insun Kang | 2193 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *09 September 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1,3-6,8,12-23,25,26,28,29 and 31-54* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-6,8,12-23,25,26,28,29 and 31-54* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *9/9/2005*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

1. This action is in response to the RCE amendment filed 9/9/2005.

2. Claims 1, 3-6, 8, 12-23, 25, 26,28, 29, and 31-54 are pending in the application.

### *Specification*

3. The specification is objected to because: The auxiliary verb "may" used throughout the specification is not specific and clear enough concerning the invention's function. It is unclear whether the invention performs the described functionality or not. It should be stated in a more definitive manner. *Note: The applicant refuses to correct the objection to the specification because the portion of the cited MPEP is only directed to the abstract. However, the examiner notes that the entire specification is objected in the previous action ("The specification is objected to...throughout the specification") and the MPEP portion cited is for reference only, specifically for the abstract, which is a part of the specification. Therefore, the objection to the specification is maintained.

### *Double Patenting*

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

5. Claims 1, 3-6, 8, 12-23, 25, 26,28, 29, and 31-54 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 285-381 of copending Application No. 09/518492 in view of MathWorks ("Stateflow for State Diagram Modeling User's Guide," version 4, 1997-2001).

Although the conflicting claims are not identical, they are not patentably distinct from each other because they are directed to substantially the same invention and recites only obvious differences which would have been obvious to one of ordinary skill in the art of program development at the time of invention in view of MathWorks such as simply (i) omitting/adding steps or elements along with their functions, and/or (ii) implementing the method steps with means for performing the steps, and/or (iii) computer program implementation of the method, and/or (iv) implementing a system, product and medium for performing the method steps. '492 is a non-published application, therefore, the specific claim languages are not stated at this time.

This is a provisional obviousness-type double patenting rejection.

6. Claims 1, 3-6, 8, 12-23, 25, 26,28, 29, and 31-54 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over U.S. Patent No. 7,000,190 in view of MathWorks ("Stateflow for State Diagram Modeling User's Guide," version 4, 1997-2001).

Although the conflicting claims are not identical, they are not patentably distinct from each other because they are directed to substantially the same invention and recites only obvious differences which would have been obvious to one of ordinary skill in the art of program development at the time of invention such as simply (i) omitting/adding steps or elements along with their functions, and/or (ii) implementing the method steps with means for performing the steps, and/or (iii) computer program implementation of the method, and/or (iv) implementing a system, product and medium for performing the method steps, as explained below.

The following example is given:

Per claim 25:

Patent '190 recites A computer-implemented method for automatically generating a graphical program ("A computer-implemented method for programmatically modifying a graphical program"); displaying an initial state diagram ("the GPG program receiving information, wherein the information specifies desired functionality of the graphical program"). '492 does not explicitly recite the user input is a state diagram information. However, MathWorks teaches that receiving state diagram information was known in the pertinent art, at the time applicant's invention was made, to describe the behavior of a system ("Stateflow...visually model and simulate complex reactive systems based on finite state machine theory," page 1-2; page 2-2) such as that disclosed in MathWorks. It would have been obvious for one having ordinary skill in the art to modify '455's disclosed system to incorporate the teachings of MathWorks. The modification would be

obvious because one having ordinary skill in the art would be motivated to represent a complex task in a visual model as suggested by MathWorks.

- automatically generating a graphical program corresponding to the initial state diagram, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program wherein the graphical program is executable by a compute, wherein said automatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating; receiving user input specifying a change to the initial state diagram; automatically updating the graphical program to correspond to the specified change, in response to the user input specifying the change, wherein said automatically updating the graphical program updates the graphical program without any user input specifying the graphical program during said updating ("programmatically modifying the graphical program in response to said information specifying the desired functionality of the graphical program, such that the graphical program implements the specified desired functionality; wherein the graphical program comprises a flow diagram comprising a plurality of interconnected nodes that visually indicate the functionality of the graphical program, wherein the graphical program is executable to perform said functionality according to the flow diagram; wherein said programmatically modifying the graphical program is performed automatically without any user input specifying the modification) as claimed.

***Claim Rejections - 35 USC § 102***

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1, 3-6, 8,12-23, 25, 26,28, 29, 31-34, and 36-54 are rejected under 35 U.S.C. 102(b) as being anticipated by MathWorks ("Stateflow for State Diagram Modeling User's Guide," version 4, 1997-2001).

Per claim 1:

MathWorks discloses:

-receiving the state diagram information, wherein the state diagram information represents the state diagram and specifies a plurality of states ("Stateflow is used together with Simulink ...Simulink supports development ...in a graphical block diagram environment," page 1-3; "A Stateflow diagram is a graphical representation of a finite state machine where states and transitions form the basic building blocks of the system...Stateflow provides a block that you include in a Simulink model," page 2-2)

-automatically generating the graphical program in response to the state diagram information ("creating a Simulink model with a Stateflow block," page 1-6)

-wherein said automatically generating comprises automatically generating graphical source code corresponding to the plurality of states, wherein the graphical source code comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program ("A Simulink model can consist of combinations of Simulink blocks, toolbox blocks, and Stateflow blocks," page 2-4; see the figure in page 2-7)

-wherein the graphical program is executable by a computer (The Simulink model and Stateflow machine work seamlessly together. Running a simulation automatically executes both the Simulink and Stateflow portions of the model," page 2-4)
-said automatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating (see the figure in section Creating a Simulink Model, page 1-6; "By default, an untitled Simulink model with an untitled, empty Stateflow block is created for you when you open the Stateflow model window. You can either start with the default empty model...to include a Stateflow diagram in an existing Simulink model," page 1-6) as claimed.

Per claims 3-6:

A state diagram is used to describe the behavior of a system and each diagram usually represents objects of an individual class and identifies the different states of its objects through the system. As an algorithm is any sequence of operations for performing a specific task, the state diagram can represent any desired operations, any other non-software system so that each state of operation can be specified, conceptualized, visualized, and constructed in the diagram. Thus, a state diagram can represent desired operation of a software program, a hardware device, algorithm, and test sequence. Therefore, accordingly, MathWorks anticipate these claims.

Per claim 8:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

- automatically generating a block diagram including the graphical source code corresponding to the specified plurality of states ("creating a Simulink model with a Stateflow block," page 1-6) as claimed.

Per claim 12:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

- for at least one state of the plurality of states, the state diagram information specifies program code associated with the state; wherein the automatically generated graphical source code includes the specified program code ("creating a Simulink model with a Stateflow block," page 1-6) as claimed.

Per claim 13:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

- for at least one state, the state diagram information specifies program code associated with the state; wherein the automatically generated graphical source code is operable to invoke the specified source code (see the figure in section Creating a Simulink Model, page 1-6) as claimed.

Per claim 14:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

- the state diagram information further specifies one or more state transitions, wherein each state transition specifies a transition from a first state to a second state; wherein

said automatically generating further comprises automatically generating graphical source code corresponding to the specified state transitions (See the section Creating a Stateflow diagram, 4 Create transitions, page 1-10 and 1-11) as claimed.

Per claim 15:

The rejection of claim 14 is incorporated, and further, MathWorks teaches:

-the automatically generated graphical source code includes placeholder graphical source code for each state transition (see the figure in section Creating a Simulink Model; "You can either start with the default empty model or copy the untitled Stateflow block into any S, page 1-6) as claimed.

Per claim 16:

The rejection of claim 15 is incorporated, and further, MathWorks teaches:

-for one or more state transitions, a user manually entering graphical source code specifying a Boolean condition associated with the state transition (section Transitions in page 7-14-7-26) as claimed.

Per claim 17:

The rejection of claim 14 is incorporated, and further, MathWorks teaches:

-wherein the state diagram information specifies at least two state transitions from a particular state; wherein the state diagram information also specifies a priority ordering for the at least two state transitions; wherein said automatically generating comprises automatically generating graphical source code such that, during execution of the

graphical program, Boolean conditions associated with the at least two state transitions are evaluated in the specified priority ordering (section Transitions in page 7-14-7-26) as claimed.

Per claim 18:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

- the state diagram information further specifies an initially active state; wherein said automatically generating comprises automatically generating graphical source code such that the graphical program begins execution in the initially active state (see section Creating and Changing States, page 3-15-3-21) as claimed.

Per claim 19:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

- the state diagram information further specifies one or more stop states; wherein said automatically generating comprises automatically generating graphical source code such that if during execution of the graphical program one of the stop states becomes active, then the graphical program is caused to stop execution (see section Creating and Changing States, page 3-15-3-21) as claimed.

Per claim 20:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

- receiving information specifying a change to the state diagram information; automatically updating the graphical program to reflect the specified change (see

section Creating and Changing States, page 3-15-3-21; Inputting Events from Simulink, page 5-16) as claimed.

Per claim 21:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

-calling an application programming interface (API) enabling the programmatic generation of a graphical program (see API properties and methods in Appendices A-C) as claimed.

Per claim 22:

The rejection of claim 1 is incorporated, and further, MathWorks teaches:

-automatically requesting a server program to generate the graphical program (see API properties and methods in Appendices A-C) as claimed.

Per claims 23 and 24, they are another method versions of claims 1 and 7, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1 and 7 above.

Per claim 25, this claim is another version of the claimed method discussed in claim 20, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth the above.

Per claims 26-28, they are the system versions of claims 1, 7, and 8, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1, 7, and 8 above.

Per claims 29-31, they are the memory medium versions of claims 1, 7, and 8, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1, 7, and 8 above.

Per claims 32-34, these claims are another versions of the claimed method discussed in claims 1, 16, and 18, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth the above.

Per claim 36:

MathWorks discloses:

-receiving the state diagram information, wherein the state diagram information represents the state diagram and specifies a plurality of states ("Stateflow is used together with Simulink ...Simulink supports development ...in a graphical block diagram environment," page 1-3; "A Stateflow diagram is a graphical representation of a finite state machine where states and transitions form the basic building blocks of the system...Stateflow provides a block that you include in a Simulink model," page 2-2)

-automatically generating the graphical program in response to the state diagram information ("creating a Simulink model with a Stateflow block," page 1-6)

-wherein said automatically generating comprises automatically generating graphical source code corresponding to the plurality of states, wherein the graphical source code comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program ("A Simulink model can consist of combinations of Simulink blocks, toolbox blocks, and Stateflow blocks," page 2-4; see the figure in page 2-7)

-wherein the graphical program is executable by a computer (The Simulink model and Stateflow machine work seamlessly together. Running a simulation automatically executes both the Simulink and Stateflow portions of the model," page 2-4)
-said automatically generating the graphical program creates the graphical program without any user input selecting the nodes or establishing connections between the nodes (see the figure in section Creating a Simulink Model, page 1-6) as claimed.

Per claims 37 and 38, they are the memory medium versions of claims 36 respectively, and are rejected for the same reasons set forth in connection with the rejection of claim 36 above.

Per claims 39-48, they are the memory medium versions of claims 8-20 respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 8-20 above.

Per claim 49:

MathWorks discloses:

The rejection of claim 37 is incorporated, and further, MathWorks teaches:
-the state diagram information comprises an executable program (The Simulink model and Stateflow machine work seamlessly together. Running a simulation automatically executes both the Simulink and Stateflow portions of the model," page 2-4) as claimed.

Per claim 50, this claim is the method version of the claimed medium discussed in claim 37, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth the above.

Per claims 51 and 52, these claims are another version of the claimed method discussed in claim 37 and 38, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth the above.

Per claims 53 and 54, these claims are another version of the claimed method discussed in claim 37 and 38, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth the above.

### *Claim Rejections - 35 USC § 103*

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claim 35 is rejected under 35 U.S.C. 103(a) as being unpatentable over MathWorks ("Stateflow for State Diagram Modeling User's Guide," version 4, 1997-2001) in view of Kodosky et al. (US 5,732,277).

Per claim 35, MathWorks does not explicitly teach that the placeholder graphical source code for each state comprises a case in a graphical case structure. However, Kodosky et al. disclose that the placeholder graphical source code for each state comprises a case in a graphical case structure (col 20, lines 30-49;col 11, lines 43-60, col 11, lines

44-60) so that it is easy for a user to cycle through the alternatives of each case. Therefore, It would have been obvious to one having ordinary skill in the art at the time of the invention was made to incorporate the teaching of Kodosky et al. to the system of MathWorks. The modification would have been obvious because one having ordinary skill in the art would have been motivated to include a case structure so that a menu list of alternatives on the screen for a user to choose from is available.

***Response to Amendment***

11. In claim 23 of the amendment, the deleted word, "automatically" in line 11 shown by strike-through was not presented in the original version. Clarification is requested.

The new abstract (page 3) must be submitted on a separate sheet (37 CFR 1.72).

***Response to Arguments***

12. Applicant's arguments filed 1/18/2005 have been fully considered but they are not persuasive.

Per claim 1:

The applicant states that:

The cited portions of the Stateflow manual clearly describe manual inclusion of a Stateflow block in a Simullink model diagram...the Stateflow block is used as an element in the Simulink model, and is not used as a basis for automatically or automatically generating graphical program source code. Nowhere

does MathWorks teach or describe automatically or automatically generating graphical program source code based on received state diagram information (page 27).

In response, "automatically generating" is interpreted to refer to an action being performed by either a program or the user and MathWork clearly recites that a Stateflow diagram is a graphical representation of a finite state machine where states and transitions form the basic building blocks of the system...Stateflow provides a block that you include in a Simulink model ( page 2-2)" in a graphical block diagram environment . Furtermore, MathWorks states "Stateflow Coder generates...code based on the Stateflow machine (page 1-4)." Therefore, MathWorks discloses automatic generation of graphical program source code based on state diagram information. If applicant means anything more, this must be brought out in the claims to further clarify the invention. Accordingly, in view of the broadest reasonable interpretation, MathWorks discloses the limitations in claim 1; therefore, the rejection of claim 1 is considered proper and maintained.

Per claims 23, 25, 26, and 29:

The applicant states that MathWorks does not disclose the limitations of claims 23, 25, 26, and 29, for the reasons set forth in connection with claim 1. As shown above, the rejection of claim 1 by MathWorks was maintained, and accordingly, the rejections of claims 23, 25, 26, and 29 are also maintained.

Per claim 32:

In response to applicant's argument that the reference fails to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., the second one or more nodes correspond to the framework embodiments...in which basic structure of the graphical program is programmatically or automatically generated, but source code for these nodes is provided or configured by the user, page 27) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). As such, the claims are read with the broadest reasonable interpretation in mind (Note MPEP 2111).

Per claim 35:

The applicant states that:

Nothing disclosed in either of the cited references provides or suggests a motivation to combine the references. Thus Applicant submits that the attempted combination of MathWorks and Kodosky is improper. Additionally, Applicant submits that even were MathWorks and Kodosky properly combinable, which Applicant argues they are not, the resulting combination would still not teach Applicant's invention as represented in claim 35. For example, neither reference discloses or even hints at the programmatic or automatic generation of graphical program sourced code based on received state diagram information, and more specifically, neither reference teaches or describes: wherein a second one or more nodes are user-configurable...comprises a case in a graphical case structure (page 30)."

In response, the applicant fails to show that the reasons to combine and motivations concerning the rejection of claim 35 are improper. Furthermore, it is noted that Kodosky uses a case structure, which is a well-known programmatic structure in

the art of programming (see "Case (Conditional) Selection Structure," col 20, lines 30-49;col 11, lines 43-60, col 11, lines 44-60). Kodosky's reference is provided as one of references that teaches the known feature. MathWorks discloses automatic generation of graphical program source code based on received state diagram information as addressed above and Kodosky teaches a case structure, Thus, all the graphical programming aspects described in MathWorks do fulfill the features brought out in applicant's claims, given that the programming aspect of Kodosky is combined into them, for which the motivation is as given above. If applicant means anything more, this must be brought out in the claims to further clarify the invention.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Insun Kang whose telephone number is 571-272-3724. The examiner can normally be reached on M-F 7:30-4 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on 571-272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

I. Kang
Examiner

WEI ZHEN
SUPERVISORY PATENT EXAMINER